postel

Network Working Group
Request for Comments: 1047

Craig Partridge
CIC at BBN Labs
February 1988

# DUPLICATE MESSAGES AND SMTP

## STATUS OF THIS MEMO

An examination of a synchronization problem in the Simple Mail Transfer Protocol (SMTP) is presented. This synchronization problem can cause a message to be delivered multiple times. A method for avoiding this problem is suggested. Nodding familiarity with the SMTP specification, RFC-821, is required. Distribution of this memo is unlimited.

## INTRODUCTION

Over the past few years, the staff of the CSNET Coordination and Information Center (CIC) has often been asked to help determine why a single mail message is being delivered multiple times to its recipients. In the process of tracing the problems of multiple delivery, we have discovered that many duplicate messages are the result of a synchronization problem in SMTP. There is a point in the process of delivering a message where the receiving mailer knows it has accepted the message but the sending mailer is still not sure the message has been reliably delivered. If the SMTP conversation is broken at this point, the sending mailer will be forced to re-deliver the message, even though the message has already been received and delivered by the receiving mailer.

## DESCRIPTION OF THE PROBLEM

The synchronization problem occurs at the end of delivering a message. When the sending mailer has finished sending the text of a message, it is required to send a line containing a single dot or period. When the receiving mailer receives this final dot, it is expected to do its final message processing and either confirm receipt of the message (with a 250 reply) or reject the message with any one of several error codes.

Observe that there is a potential synchronization gap here. During the period between the time the receiving mailer has determined that it will accept the message, and the time that sending mailer gets the 250 reply, the message is active at both the sending and receiving mailer. Until the sending mailer gets the 250 reply, it must assume the message was not delivered. After the receiving mailer has

decided to accept the message, it must assume the message has been delivered to it. If the communications link fails during this synchronization gap, then the message has been duplicated. Both mailers have active copies of the message that they will try to deliver.

It may be hard to believe that this problem is the cause of many duplicate messages. Intuitively, one might expect that the time spent in the state between the final dot and its accepting 250 reply is quite small. In practice, however, this period is often quite long; long enough that timeouts by the sending mailer (or possibly network failures) are quite common. Observations by the author suggest that this synchronization problem may be the second leading cause of duplicate messages on the Internet (second to mail loops).

Many mailers delay responding to the final dot because they are doing sophisticated processing of the message, in an attempt to confirm that they can deliver the message. For example, the mailers may expand an entire mailing list to confirm that it can reach all addressees or may attempt to physically deposit the message into the mailboxes of local users, before confirming receipt of the final dot. These practices are not unreasonable, but they often cause the synchronization gap to continue for several minutes, and increase the likelihood that the sending mailer will timeout or the network will fail before the accepting 250 reply is sent.

## AVOIDING SYNCHRONIZATION PROBLEMS

The best way to avoid the synchronization problem is to minimize the length of the synchronization gap. In other words, receiving mailers should acknowledge the final dot as soon as possible and do more complex processing of the message later.

RFC-821 (on page 22) states that unless the receiving mailer is completely unable to process a message it should accept the message and acknowledge any errors in processing in a separate message or messages sent back to the originator of the message. As a result, receiving mailers should be able to acknowledge the final dot as soon as the message has been safely put in a non-volatile (e.g., disk) queue for further processing. Fast acceptance of a message does not violate RFC-821.

Some mailers can be configured to do more or less processing upon receipt of the final dot. In such situations, the mailer should always be configured to do less processing.

Finally, some mailers allow remote mailers only a minute or two to acknowledge the final dot before timing out and trying again. Given

the increasing round-trip times on the Internet, and that some processing after the final dot is required, the timeout for reply to the final dot should probably be at least 5 minutes and a timeout of 10 minutes would not be unreasonable.